Generic Attacks on Feistel Schemes

Jacques Patarin[1,2]

[1] CP8 Crypto Lab, SchlumbergerSema, 36-38 rue de la Princesse,
BP 45, 78430 Louveciennes Cedex, France
[2] PRiSM, University of Versailles, 45 av. des États-Unis,
78035 Versailles Cedex, France

Abstract. Let A be a Feistel scheme with 5 rounds from $2n$ bits to $2n$ bits. In the present paper we show that for most such schemes A:

1. It is possible to distinguish A from a random permutation from $2n$ bits to $2n$ bits after doing at most $\mathcal{O}(2^{\frac{7n}{4}})$ computations with $\mathcal{O}(2^{\frac{7n}{4}})$ **random** plaintext/ciphertext pairs.
2. It is possible to distinguish A from a random permutation from $2n$ bits to $2n$ bits after doing at most $\mathcal{O}(2^{\frac{3n}{2}})$ computations with $\mathcal{O}(2^{\frac{3n}{2}})$ **chosen** plaintexts.

Since the complexities are smaller than the number 2^{2n} of possible inputs, they show that some generic attacks always exist on Feistel schemes with 5 rounds. Therefore we recommend in Cryptography to use Feistel schemes with at least 6 rounds in the design of pseudo-random permutations.

We will also show in this paper that it is possible to distinguish most of 6 round Feistel permutations generator from a truly random permutation generator by using a few (i.e. $\mathcal{O}(1)$) permutations of the generator and by using a total number of $\mathcal{O}(2^{2n})$ queries and a total of $\mathcal{O}(2^{2n})$ computations. This result is not really useful to attack a single 6 round Feistel permutation, but it shows that when we have to generate several pseudo-random permutations on a small number of bits we recommend to use more than 6 rounds. We also show that it is also possible to extend these results to any number of rounds, however with an even larger complexity.

Keywords: Feistel permutations, pseudo-random permutations, generic attacks on encryption schemes, Luby-Rackoff theory.

1 Introduction

Many secret key algorithms used in cryptography are Feistel schemes (a precise definition of a Feistel scheme is given in section 2), for example DES, TDES, many AES candidates, etc.. In order to be as fast as possible, it is interesting to have not too many rounds. However, for security reasons it is important to have a sufficient number of rounds. Generally, when a Feistel scheme is designed for cryptography, the designer either uses many (say ≥ 16 as in DES) very simple rounds, or uses very few (for example 8 as in DFC) more complex rounds. A natural question is: what is the minimum number of rounds required in a Feistel scheme to avoid all the "generic attacks" , i.e. all the attacks effective against most of the schemes, and with a complexity negligible compared with a search on all the possible inputs of the permutation.

Let assume that we have a permutation from $2n$ bits to $2n$ bits. Then a generic attack will be an attack with a complexity negligible compared to $\mathcal{O}(2^{2n})$, since there are 2^{2n} possible inputs on $2n$ bits.

It is easy to see that for a Feistel scheme with only one round there is a generic attack with only 1 query of the permutation and $\mathcal{O}(1)$ computations: just check if the first half (n bits) of the output are equal to the second half of the input.
In [4] it was shown that for a Feistel scheme with two rounds there is also a generic attack with a complexity of $\mathcal{O}(1)$ chosen inputs (or $\mathcal{O}(2^{\frac{n}{2}})$ random inputs).
Also in [4], M. Luby and C. Rackoff have shown their famous result: for more than 3 rounds all generic attacks on Feistel schemes require at least $\mathcal{O}(2^{\frac{n}{2}})$ inputs, even for chosen inputs. If we call a Luby-Rackoff construction (a.k.a. L-R construction) a Feistel scheme instantiated with pseudo-random functions, this result says that the Luby-Rackoff construction with 3 rounds is a pseudorandom permutation.

Moreover for 4 rounds all the generic attacks on Feistel schemes require at least $\mathcal{O}(2^{\frac{n}{2}})$ inputs, even for a stronger attack that combines chosen inputs and chosen outputs (see [4] and a proof in [6], that shows that the Luby-Rackoff construction with 4 rounds is super-pseudorandom, a.k.a strong pseudorandom). However it was discovered in [7] (and independently in [1]) that these lower bounds on 3 and 4 rounds are tight, i.e. there exist a generic attack on all Feistel schemes with 3 or 4 rounds with $\mathcal{O}(2^{\frac{n}{2}})$ chosen inputs with $\mathcal{O}(2^{\frac{n}{2}})$ computations.

For 5 rounds or more the question remained open. In [7] it was proved that for 5 rounds (or more) the number of queries must be at least $\mathcal{O}(2^{\frac{2n}{3}})$ (even with unbounded computation complexity), and in [8] it was shown that for 6 rounds (or more) the number of queries must be at least $\mathcal{O}(2^{\frac{3n}{4}})$ (even with unbounded computations).

It can be noticed (see [7]) that if we have access to unbounded computations, then we can make an exhaustive search on all the possible round functions of the Feistel scheme, and this will give an attack with only $\mathcal{O}(2^n)$ queries (see [7]) but a gigantic complexity $\geq \mathcal{O}(2^{n2^n})$. This "exhaustive search" attack always exists, but since the complexity is far much larger than the exhaustive search on plaintexts in $\mathcal{O}(2^{2n})$, it was still an open problem to know if generic attacks, with a complexity $\ll \mathcal{O}(2^{2n})$, exist on 5 rounds (or more) of Feistel schemes.

In this paper we will indeed show that there exist generic attacks on 5 rounds of the Feistel scheme, with a complexity $\ll \mathcal{O}(2^{2n})$. We describe two attacks on 5 round Feistel schemes:

1. An attack with $\mathcal{O}(2^{\frac{7n}{4}})$ computations on $\mathcal{O}(2^{\frac{7n}{4}})$ **random** input/output pairs.
2. An attack with $\mathcal{O}(2^{\frac{3n}{2}})$ computations on $\mathcal{O}(2^{\frac{3n}{2}})$ **chosen** inputs.

For 6 rounds (or more) the problem remains open. In this paper we will describe some attacks on 6 rounds (or more) with a complexity much smaller than $\mathcal{O}(2^{n2^n})$ of exhaustive search, but still $\geq \mathcal{O}(2^{2n})$. So these attacks on 6 rounds and more are generally not interesting against a single permutation. However they may be useful when several permutations are used, i.e. they will be able to distinguish some permutation generators. These attacks show for example that when several small permutations must be generated (for example in the Graph Isomorphism scheme, or as in the Permuted Kernel scheme) then we must not use a 6 round Feistel construction.

Remark The generic attacks presented here for 3, 4 and 5 rounds are effective against most Feistel schemes, or when the round functions are randomly chosen. However it can occur that for specific choices of the round function, the attacks, performed exactly as described, may fail. However in this case, very often there are modified attacks on these specific round functions. This point will be discussed in section 6.

2 Notations

We use the following notations that are very similar to those used in [4], [5] and [8].

- $I_n = \{0,1\}^n$ is the set of the 2^n binary strings of length n.
- For $a, b \in I_n$, $[a,b]$ will be the string of length $2n$ of I_{2n} which is the concatenation of a and b.
- For $a, b \in I_n$, $a \oplus b$ stands for bit by bit exclusive or of a and b.
- $\circ$ is the composition of functions.
- The set of all functions from I_n to I_n is F_n. Thus $|F_n| = 2^{n \cdot 2^n}$.
- The set of all permutations from I_n to I_n is B_n. Thus $B_n \subset F_n$, and $|B_n| = (2^n)!$
- Let f_1 be a function of F_n. Let L, R, S and T be elements of I_n. Then by definition

$$\Psi(f_1)[L,R] = [S,T] \overset{\text{def}}{\Leftrightarrow} \begin{cases} S = R \\ \text{and} \\ T = L \oplus f_1(R) \end{cases}$$

- Let $f_1, f_2, \ldots, f_k$ be k functions of F_n. Then by definition:

$$\Psi^k(f_1, \ldots, f_k) = \Psi(f_k) \circ \cdots \circ \Psi(f_2) \circ \Psi(f_1).$$

The permutation $\Psi^k(f_1, \ldots, f_k)$ is called “a Feistel scheme with k rounds” and also called Ψ^k.

3 Generic attacks on 1,2,3 and 4 rounds

Up till now, generic attacks had been discovered for Feistel schemes with 1,2,3,4 rounds. Let us shortly describe these attacks.

Let f be a permutation of B_{2n}. For a value $[L_i, R_i] \in I_{2n}$ we will denote by $[S_i, T_i] = f[L_i, R_i]$.

1 round

The attack just tests if $S_1 = R_1$. If f is a Feistel scheme with 1 round, this will happen with 100% probability, and if f is a random permutation with probability $\simeq \frac{1}{2^n}$. So with one round there is a generic attack with only 1 random query and $\mathcal{O}(1)$ computations.

2 rounds

Let choose $R_2 = R_1$ and $L_2 \neq L_1$. Then the attack just tests if $S_1 \oplus S_2 = L_1 \oplus L_2$. This will occur with 100% probability if f is a Feistel scheme with 2 rounds, and if f is a random permutation with probability $\simeq \frac{1}{2^n}$. So with two rounds there is a generic attack with only 2 chosen queries and $\mathcal{O}(1)$ computations.

Note 1: It is possible to transform this chosen plaintext attack in a known plaintext attack like the following. If we have $\mathcal{O}(2^{\frac{n}{2}})$ random inputs $[L_i, R_i]$, then with a good probability we will have a collision $R_i = R_j, i \neq j$. Then we test if $S_i \oplus S_j = L_i \oplus L_j$. Now the attack requires $\mathcal{O}(2^{\frac{n}{2}})$ random queries and $\mathcal{O}(2^{\frac{n}{2}})$ computations.

Note 2: This attack on 1 and 2 rounds was already described in [4].

3 rounds

Let ϕ be the following algorithm :

1. ϕ chooses m distinct $R_i, 1 \leq i \leq m$, and chooses $L_i = 0$ (or L_i constant) for all i, $1 \leq i \leq m$.
2. ϕ asks for the values $[S_i, T_i] = f[L_i, R_i], 1 \leq i \leq m$.
3. ϕ counts the number N of equalities of the form $R_i \oplus S_i = R_j \oplus S_j, i < j$.
4. Let N_0 be the expected value of N when f is a random permutation, and N_1 be the expected value of N when f is a $\psi^3(f_1, f_2, f_3)$, with randomly chosen f_1, f_2, f_3.
   Then $N_1 \simeq 2N_0$, because when f is a $\psi^3(f_1, f_2, f_3)$, $R_i \oplus S_i = f_2(f_1(R_i))$ so $f_2(f_1(R_i)) = f_2(f_1(R_j)), i < j$, if $f_1(R_i) \neq f_1(R_j)$ and $f_2(f_1(R_i)) = f_2(f_1(R_j))$ <u>or</u> if $f_1(R_i) = f_1(R_j)$.

So by counting N we will obtain a way to distinguish 3 round Feistel permutations from random permutations. This generic attack requires $\mathcal{O}(2^{\frac{n}{2}})$ chosen queries and $\mathcal{O}(2^{\frac{n}{2}})$ computations (just store the values $R_i \oplus S_i$ and count the collisions).

Remark Here $N_1 \simeq 2 \cdot N_0$ when f_1, f_2, f_3 are randomly chosen. Therefore this attack is effective on most of 3 round Feistel schemes but not necessarily on all 3 round Feistel schemes. (See section 6 for more comments on this point).

4 rounds

This time, we take $R_i = 0$ (or R_i constant), and we count the number N of equalities of the form $S_i \oplus L_i = S_j \oplus L_j$, $i < j$. In fact, when $f = \psi^4(f_1, f_2, f_3, f_4)$, then $S_i \oplus L_i = f_3(f_2(L_i \oplus f_1(0))) \oplus f_1(0)$. So the probability of such an equality is about the double in this case (as long as f_1, f_2, f_3 are randomly chosen) than in the case where f is a random permutation (because if $f_2(L_i \oplus f_1(0)) = f_2(L_j \oplus f_1(0))$ this equality holds, and if $\beta_i = f_2(L_i \oplus f_1(0)) \neq f_2(L_j \oplus f_1(0)) = \beta_j$ but $f_3(\beta_i) = f_3(\beta_j)$, this equality also holds).

So by counting N we will obtain a way to distinguish 4 round Feistel permutations from random permutations. This generic attack requires $\mathcal{O}(2^{\frac{n}{2}})$ chosen queries and $\mathcal{O}(2^{\frac{n}{2}})$ computations (just store the values $S_i \oplus L_i$ and count the collisions).

Notes:

1. These attacks for 3 and 4 rounds have been first published in [7], and independently re-discovered in [1].
2. Here again the attack is effective against most of 4 round Feistel schemes but not necessarily on all 4 round Feistel schemes. (See section 6 for more comments on this point).

4 A generic attack on 5 round Feistel permutations with $\mathcal{O}(2^{\frac{7n}{4}})$ random plaintexts and $\mathcal{O}(2^{\frac{7n}{4}})$ complexity

4.1 Notations for 5 round Feistel permutations

Let i be an integer. For any given i, let $[L_i, R_i]$ be a string of $2n$ bits in I_{2n}. Let

$$\Psi^5[L_i, R_i] = [S_i, T_i].$$

We introduce the intermediate variables X_i, P_i and Y_i such that:

$$\begin{cases} X_i = L_i \oplus f_1(R_i) \\ P_i = R_i \oplus f_2(X_i) \\ Y_i = X_i \oplus f_3(P_i) \end{cases}$$

So we have: $S_i = P_i \oplus f_4(Y_i)$ and $T_i = Y_i \oplus f_5(S_i)$. In other terms we have the following:

$$\begin{aligned} \Psi(f_1)[L_i, R_i] &= [R_i, X_i], \text{ as } X_i = L_i \oplus f_1(R_i) \\ \Psi(f_2)[R_i, X_i] &= [X_i, P_i], \text{ as } P_i = R_i \oplus f_2(X_i) \\ \Psi(f_3)[X_i, P_i] &= [P_i, Y_i], \text{ as } Y_i = X_i \oplus f_3(P_i) \\ \Psi(f_4)[P_i, Y_i] &= [Y_i, S_i], \text{ as } S_i = P_i \oplus f_4(Y_i) \\ \Psi(f_5)[Y_i, S_i] &= [S_i, T_i], \text{ as } T_i = Y_i \oplus f_5(S_i) \end{aligned}$$

Input:	L	R
1 round:	R	X
2 rounds:	X	P
3 rounds:	P	Y
4 rounds:	Y	S
Output, 5 rounds:	S	T

Figure 1.

We may notice that the following conditions (C) are always satisfied:

$$(\mathbf{C})\begin{cases} R_i = R_j \Rightarrow X_i \oplus L_i = X_j \oplus L_j & \mathbf{(CR)} \\ X_i = X_j \Rightarrow R_i \oplus P_i = R_j \oplus P_j & \mathbf{(CX)} \\ P_i = P_j \Rightarrow X_i \oplus Y_i = X_j \oplus Y_j & \mathbf{(CP)} \\ Y_i = Y_j \Rightarrow S_i \oplus P_i = S_j \oplus P_j & \mathbf{(CY)} \\ S_i = S_j \Rightarrow Y_i \oplus T_i = Y_j \oplus T_j & \mathbf{(CS)} \end{cases}$$

4.2 The attack

Let f be a permutation from B_{2n} We want to know (with a good probability) if f is a random element of B_{2n}, or if f is a Feistel scheme with 5 rounds (i.e. $f = \Phi^5(f_1, f_2, f_3, f_4, f_5)$ with f_1, f_2, f_3, f_4, f_5 being 5 functions of F_n).

The attack proceeds as follows:

Step 1: We generate m values $[S_i, T_i] = f[L_i, R_i]$, $1 \le i \le m$ such that the $[L_i, R_i]$ values are randomly chosen in I_{2n} and with $m = \mathcal{O}(2^{\frac{7n}{4}})$.

Step 2: We look if among these values, we can find 4 pairwise distinct indices denoted by $1, 2, 3, 4$ such that the following 8 equations (and 2 inequalities) are satisfied:

$$(\#)\begin{cases} R_1 = R_3 \\ R_2 = R_4 \\ L_1 \oplus L_3 = L_2 \oplus L_4 \\ S_1 = S_3 \\ S_2 = S_4 \\ S_1 \oplus S_2 = R_1 \oplus R_2 \\ T_1 \oplus T_3 = L_1 \oplus L_3 \\ T_1 \oplus T_3 = T_2 \oplus T_4 \end{cases}$$

(and with $R_1 \neq R_2$ and $L_1 \neq L_3$)

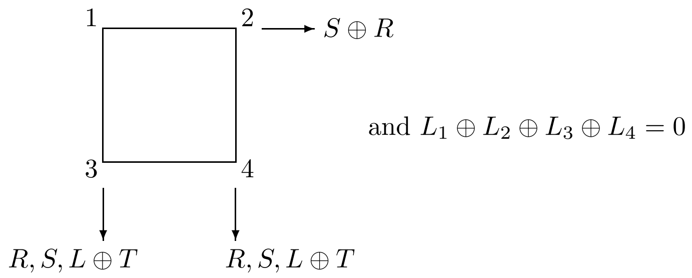

Figure 2: A representation of the 8 *equations* # *in* L, S, R, T.

Below we explain how one can test with the complexity of $\mathcal{O}(m)$ if such indices exist.

Step 3: If such indices exist, we will guess that f is Feistel scheme with 5 rounds. If not we will say that f is not a Feistel scheme. [We will see below that the probability to find such indices is not negligible if f is a Feistel scheme with 5 rounds and $M \geq \mathcal{O}(2^{\frac{7n}{4}})$ for most of 5 round Feistel schemes].

4.3 How to accomplish the step 2 in $\mathcal{O}(m)$ computations

First, we find among the $m \times m$ possibilities, all the possible indices 1 and 3 such that:

$$\begin{cases} R_1 = R_3 \\ S_1 = S_3 \\ L_1 \oplus T_1 = L_3 \oplus T_3 \end{cases}$$

It is possible to this in $\mathcal{O}(m)$ computations instead of $\mathcal{O}(m^2)$ by storing all the m values $(R_i, S_i, L_i \oplus T_i)$ in a hash table and looking for collisions. We expect to find $\frac{m^2}{2^{3n}} \ll m$ such indices (as $m \ll 2^{3n}$).

In the same way we find all the possible indices 2 and 4 such that:

$$\begin{cases} R_2 = R_4 \\ S_2 = S_4 \\ L_2 \oplus T_2 = L_4 \oplus T_4 \end{cases}$$

Each part requires $\mathcal{O}(m)$ computations and $\mathcal{O}(m)$ of memory, and, if needed, there is a tradeoff with $\mathcal{O}(m \cdot \alpha)$ computations and $\mathcal{O}(m/\alpha)$ memory.

Now we store all the values $(L_1 \oplus L_3, S_1 \oplus R_1)$ for all the indices $(1,3)$ already found. There are about $\frac{m^2}{2^{3n}} \leq m$ such values. Then we store all the values $(L_2 \oplus L_4, S_2 \oplus R_2)$ for all the indices $(2,4)$ already found. Using another birthday paradox technique, we look for the following collision:

$$\begin{cases} L_2 \oplus L_4 = L_1 \oplus L_3 \\ S_2 \oplus R_2 = S_1 \oplus R_1 \end{cases}$$

The complexity and the storage is $\mathcal{O}(\frac{m^2}{2^{3n}}) \leq \mathcal{O}(m)$ again. At the end we have at most m choices of pairwise distinct indices $(1,2,3,4)$. Among these we keep those that give $R_1 \neq R_2$ and $L_1 \neq L_3$. By inspection we check that now they satisfy all the equations of (#).

4.4 Probability of (#) when f is a random permutation of B_{2n}

When f is a random permutation of B_{2n}, we have $\mathcal{O}(m^4)$ possibilities to chose the indices $1,2,3,4$ among the m possible indices, and we have 8 equations to satisfy, with a probability about $\frac{1}{2^{8n}}$ to have them all true for some pairwise distinct $1,2,3,4$. By inspection we check that the equations of (#) are not dependent. Thus the probability to have 4 pairwise distinct indices $1,2,3,4$ that satisfy (#) is about $\frac{m^4}{2^{8n}}$ when f is a random permutation of B_{2n} (n.b. the two additional inequalities $R_1 \neq R_2$ and $L_1 \neq L_3$ change nothing). Since $m \ll 2^{2n}$ (because $m = \mathcal{O}(2^{\frac{7n}{4}})$) this probability is negligible.

4.5 Probability of (#) when f is a Feistel scheme with 5 rounds

Theorem 1 *When f is a Feistel scheme with 5 rounds, the* 8 *equations of* (#) *are a logical consequence on the following* 7 *equations:*

$$(\mathbf{\Lambda})\begin{cases} R_1 = R_3 & (1)\\ R_2 = R_4 & (2)\\ L_1 \oplus L_3 = L_2 \oplus L_4 & (3)\\ S_1 = S_3 & (4)\\ X_1 = X_2 & (5)\\ P_1 = P_3 & (6)\\ Y_1 = Y_2 & (7)\end{cases}$$

Proof of Theorem 1.

We will use the facts (CR), (CX), (CP), (CY) and (CS) that have been introduced in section 4.1.

- From (1) and (CR) we get
  $X_3 = X_1 \oplus L_1 \oplus L_3$ (8)
- From (2) and (CR) we get $X_4 \oplus L_4 = X_2 \oplus L_2$, and then using (8), (5) and (3) we get
  $X_4 = X_3$ (9).
- From (5) and (CX) we get:
  $R_1 \oplus P_1 = R_2 \oplus P_2$ (10)
- From (9) and (CX) we get $R_4 \oplus P_4 = R_3 \oplus P_3$ and then from (10), (6), (1) and (2) we get:
  $P_4 = P_2$ (11)
- From (6) and (CP) we get $X_1 \oplus Y_1 = X_3 \oplus Y_3$ and then from (8) we get:
  $Y_3 = Y_1 \oplus L_1 \oplus L_3$ (12)
- From (11) and (CP) we get $X_2 \oplus Y_2 = X_4 \oplus Y_4$ and then from (12), (7), (9), (5) and (8) we get:
  $Y_4 = Y_3$ (13)
- From (7) and (CY) we get $S_1 \oplus P_1 = S_2 \oplus P_2$ and then from (10) we get:
  $S_1 \oplus S_2 = R_1 \oplus R_2$ (14)
- From (13) and (CY) we get $S_4 \oplus P_4 = S_3 \oplus P_3$ and then from (14), (4), (11), (6) and (10) we get:
  $S_4 = S_2$ (15)
- From (4) and (CS) we get $Y_1 \oplus T_1 = Y_3 \oplus T_3$ and then from (12) we get:
  $T_3 = T_1 \oplus L_1 \oplus L_3$ (16).
- From (15) and (CS) we get $Y_4 \oplus T_4 = Y_2 \oplus T_2$ and then from (13), (7), (12) and (16) we get:
  $T_4 \oplus T_2 = T_1 \oplus T_3$ (17)
- If $R_1 = R_2$ then because of (5) we have $L_1 = L_2$ and $R_1 = R_2 \Rightarrow 1 = 2$ and the indices 1 and 2 are distinct by definition. Thus
  $R_1 \neq R_2$ (18)
- Finally since $1 \neq 3$ and because of (1) we have. $L_1 \neq L_3$ (19)

So all the equations of (#) are indeed just consequences of the 7 equations (Λ) when f is a Feistel with 5 rounds. Indeed the $8+2$ conditions of (#) are now in (1), (2), (3), (4), (15), (14), (16), (17), and finally (18) and (19).

Theorem 2 *Let f be a Feistel scheme with 5 rounds, $f = \Psi^5(f_1, f_2, f_3, f_4, f_5)$. Then for most of such f, the probability to have 4 pairwise distinct indices 1,2,3,4 that satisfy # is $\geq \mathcal{O}(\frac{m^4}{2^{7n}})$, and thus is not negligible when $m \geq \mathcal{O}(2^{\frac{7n}{4}})$. Therefore the algorithm given in the section 4 is indeed a generic way to distinguish most Feistel schemes with 5 rounds from a truly random permutation of B_{2n} with a complexity of $\mathcal{O}(2^{\frac{7n}{4}})$.*

Proof.
When f_1, f_2, f_3, f_4, f_5 are randomly chosen in F_n, the probability that there exist pairwise distinct indices 1,2,3,4 chosen out of a set of m indices such that all the 7 equations (Λ) hold is $= \mathcal{O}(\frac{m^4}{2^{7n}})$. Thus from the Theorem 1 we get the Theorem 2.

Remark Here again, the attack is effective against most of 5 round Feistel schemes, but not necessarily on all 5 round Feistel schemes. (See section 6 for more comments on that).

5 A generic attack on 5 round Feistel permutations with $\mathcal{O}(2^{\frac{3n}{2}})$ chosen plaintexts and $\mathcal{O}(2^{\frac{3n}{2}})$ complexity

This attack proceeds exactly as the previous attack of the Section 4, except that now Step 1 is replaced by the following Step' 1:

Step' 1 We generate m values $f[L_i, R_i] = [S_i, T_i]$, $1 \leq i \leq m$ such that the L_i values are randomly chosen in I_n and the R_i values are randomly chosen in a subset I'_n of I_n with only $2^{\frac{n}{2}}$ elements. For example I'_n=all the strings of n bits with the first $n/2$ bits at 0.
Let $m = \mathcal{O}(2^{\frac{3n}{2}})$.

5.1 Probability of (#) when f is a random permutation of B_{2n}

Now the probability that there are some indices $1, 2, 3, 4$ such that equations (#) are satisfied when f is randomly chosen in B_{2n} is about

$$\frac{m^4}{2^{\frac{n}{2}} \cdot 2^{\frac{n}{2}} 2^{6n}} = \frac{m^4}{2^{7n}}$$

(because the equations $R_1 = R_3$ and $R_2 = R_4$ have now a probability $\frac{1}{2^{\frac{n}{2}}}$ to be satisfied instead of $\frac{1}{2^n}$).

However, since here $m = \mathcal{O}(2^{\frac{3n}{2}})$, this probability $\frac{m^4}{2^{7n}}$ is still negligible.

5.2 Probability of (#) when f is a Feistel scheme with 5 rounds

When f is a Feistel scheme with 5 rounds, with f_1, f_2, f_3, f_4, f_5 randomly chosen in F_n, the probability that there exist indices $1, 2, 3, 4$ chosen out of a set of m indices, such that all the 7 equations (Λ) are satisfied is about

$$\simeq \frac{m^4}{2^{\frac{n}{2}} \cdot 2^{\frac{n}{2}} 2^{5n}} = \frac{m^4}{2^{6n}}$$

(because the equations $R_1 = R_3$ and $R_2 = R_4$ have now a probability $\frac{1}{2^{\frac{n}{2}}}$ to be satisfied instead of $\frac{1}{2^n}$).

So from Theorem 1 of section 4, we see that for these functions f the probability that there exist indices 1,2,3,4 such that all the 8 equations (and 2 inequalities) # are satisfied is here generally $\geq \mathcal{O}(\frac{m^4}{2^{6n}})$.

Thus the algorithm given in this section 5 is indeed a generic way to distinguish most Feistel schemes with 5 rounds from a truly random permutation of B_{2n}, with a complexity $\mathcal{O}(2^{\frac{3n}{2}})$ and $\mathcal{O}(2^{\frac{3n}{2}})$ chosen queries.

Remark Here again some time/memory tradeoff is possible: use $\mathcal{O}(2^{\frac{3n}{2}})$ chosen queries, $\mathcal{O}(2^{\frac{3n}{2}} \cdot \alpha)$ computations and $\mathcal{O}(2^{\frac{3n}{2}}/\alpha)$ of memory.

6 Feistel schemes with specific round functions

The problem. The generic attacks that we have presented for 3, 4 and 5 rounds are effective against most Feistel schemes, or when the round functions are randomly chosen. However it can occur that for specific choices of the round functions, these attacks, if applied exactly as described, may fail. In this cases, very often there are some other attacks, against these specific rounds functions, that are even simpler. We will illustrate this on an example pointed out by an anonymous referee of Asiacrypt'2001.

Theorem 3 (Knudsen, see [2] or [3]) *Let $[L_1, R_1]$ and $[L_2, R_2]$ be two inputs of a 5 round Feistel scheme, and let $[S_1, T_1]$ and $[S_2, T_2]$ be the outputs. Let assume that the round functions f_2 and f_3 are permutations (therefore they are* **not** *random functions of F_n). Then if $R_1 = R_2$ and $L_1 \neq L_2$ it is impossible to have simultaneously $S_1 = S_2$ and $L_1 \oplus L_2 = T_1 \oplus T_2$.*

Proof.

$R_1 = R_2 \Rightarrow X_1 \oplus X_2 = L_1 \oplus L_2$, and $S_1 = S_2 \Rightarrow Y_1 \oplus Y_2 = T_1 \oplus T_2$. Therefore if we have $L_1 \oplus L_2 = T_1 \oplus T_2$, we will have also:

$$X_1 \oplus Y_1 = X_2 \oplus Y_2.$$

Now since we have $Y_i = X_i \oplus f_3(P_i)$, we will have $f_3(P_1) = f_3(P_2)$ and since f_3 is a permutation we get $P_1 = P_2$.

Then since we have $P_i = R_i \oplus f_2[L_i \oplus f_1(R_i)]$ with $R_1 = R_2$, and since f_2 is a permutation we get

$$L_1 \oplus f_1(R_1) = L_2 \oplus f_1(R_2).$$

This is in contradiction with $R_1 = R_2$ and $L_1 \neq L_2$.

Attacks on 5 round Feistel schemes with f_2 and f_3 permutations

From the above Theorem 3 we see that our attack given in section 4 and 5 against most 5 round Feistel schemes will fail when f_2 and f_3 are permutations. Indeed, the event $R_1 = R_3, L_1 \neq L_3, S_1 = S_3$ and $L_1 \oplus L_3 = T_1 \oplus T_3$ will never occur if f_2 and f_3 are permutations. However, in such a case there is an even simpler attack that comes immediately from the Theorem 3: we can randomly get m input/output values and count the number of indices $(i, j), i < j$ such that:

$$\begin{cases} R_i = R_j \\ S_i = S_j \\ L_i \oplus L_j = T_i \oplus T_j \end{cases}$$

For a random permutation this number is $\mathcal{O}(\frac{m^2}{2^{3n}})$, and for a 5 round Feistel scheme with f_2 and f_3 being permutations, it is exactly 0.
This attack requires $\mathcal{O}(2^{\frac{3n}{2}})$ random plaintext/ciphertext pairs and $\mathcal{O}(2^{\frac{3n}{2}})$ computations.

Remark: This attack can also be extended to 6 round Feistel schemes when the round functions are permutations (or “quasi-permutations”), see [2, 3] for details.

Conclusion It was known (before the present paper) that some generic attacks on 5 round Feistel schemes exist when the round functions are permutations. This particular case is interesting since two of the former AES candidates, namely DFC and DEAL, were such Feistel schemes using permutations as round functions. (More precisely they were “quasi-permutations” in DFC). The number of rounds in these functions is however ≥ 6.
In this paper we have shown a more general result that such generic attacks exist for most of 5 round Feistel schemes (even when f_2 and f_3 are **not** permutations). It can be noticed that our attack is based on specific relations on 4 points (corresponding to 4 ciphertexts), while the previous attacks were based on specific relations on only 2 points (”impossible differentials”).

7 Attacking Feistel Generators

In this section we will describe what is an attack against a generator of permutations (and not only against a single permutation randomly generated by a generator of permutations), i.e. we will be able to study several permutations generated by the generator. Then we will evaluate the complexity of brute force attacks and we will notice that since all Feistel permutations have an even signature, it is possible to distinguish them from a random permutation in $\mathcal{O}(2^{2n})$.

Let G be a “k round Feistel Generator”, i.e. from a binary string K, G generates a k round Feistel permutation G_K of B_{2n}.
Let G' be a truly random permutation generator, i.e. from a string K, G' generates a truly random permutation G'_K of B_{2n}.

Let G'' be a truly random even permutation generator, i.e. from a string K, G'' generates a truly random permutation G''_K of A_{2n}, with A_{2n} being the group of all the permutations of B_{2n} with even signature.

We are looking for attacks that distinguish G from G', and also for attacks that will distinguish G from G''.

Adversarial model: An attacker can choose some strings $K_1, \dots K_f$, can ask for some inputs $[L_i, R_i] \in I_{2n}$, and can ask for some $G_{K_\alpha}[L_i, R_i]$ (with K_α being one of the K_i). Here the attack is more general than in the previous sections, since the attacker can have access to many different permutations generated by the same generator.

Adversarial goal: The aim of the attacker is to distinguish G from G' (or from G'') with a good probability and with a complexity as small as possible.

Brute force attacks A possible attack is the exhaustive search on the k round functions $f_1, \dots, f_k$ form I_n to I_n that have been used in the Feistel construction. This attack always exists, but since we have $2^{k \cdot n \cdot 2^n}$ possibilities for $f_1, \dots, f_k$, this attack requires about $2^{k \cdot n \cdot 2^n}$ computations (or $2^{\lceil \frac{k}{2} \rceil \cdot n \cdot 2^n}$ computations in a version “in the middle” of the attack) and about $k \cdot 2^{n-1}$ random queries[1] and only 1 permutation of the generator.

Attack by the signature

Theorem 4 *If $n \geq 2$ then all the Feistel schemes from $I_{2n} \to I_{2n}$ have an even signature.*

Proof.
Let $\sigma : I_{2n} \to I_{2n}$
$[L, R] \mapsto [R, L]$.
Let f_1 be a function of F_n.
Let $\Psi'(f_1)[L, R] = [L \oplus f_1(R), R]$.
We will show that both σ and $\Psi'(f_1)$ have an even signature, so will have $\sigma \circ \Psi'(f_1) = \Psi(f_1)$, and thus by composition, all the Feistel schemes from $I_{2n} \to I_{2n}$ have an even signature.

For σ: All the cycles have 1 or 2 elements, and we have 2^n cycles with 1 element (and an even signature), and $\frac{2^{2n}-2^n}{2}$ cycles with 2 elements. When $n \geq 2$ this number is even.

For $\Psi'(f_1)$: All the cycles have 1 or 2 elements since $\Psi'(f_1) \circ \Psi'(f_1) = Id$. Moreover the number of cycles with 2 elements is $\frac{2^n \cdot k}{2}$, with k being the number of values R such that $f_1(R) \neq 0$. So when $n \geq 2$ the signature of $\Psi'(f_1)$ is even.

Theorem 5 *Let f be a permutation of B_{2n}. Then using $\mathcal{O}(2^{2n})$ computations on the 2^{2n} input/output values of f, we can compute the signature of f.*

[1] each query divides by about 2^{2n} the number of possible $f_1, \dots, f_k$

Proof.

Just compute all the cycles c_i of f, $f = \prod_{i=1}^{\alpha} c_i$ and use the formula: signature$(f) = \prod_{i=1}^{\alpha} (-1)^{length(c_i)+1}$.

Theorem 6 *Let G be a Feistel scheme generator, then it is possible to distinguish G from a generator of truly random permutations of B_{2n} after $\mathcal{O}(2^{2n})$ computations on $\mathcal{O}(2^{2n})$ input/output values.*

Proof.

It is direct consequence of the Theorems 4 and 5 above.

Remark.

It is however probably much more difficult to distinguish G from random permutations of A_{2n}, with A_{2n} being the group of all the permutations of B_{2n} with even signature. In the next sections we will present our best attacks for this problem.

8 An attack on 6 round Feistel Generators in $\mathcal{O}(2^{2n})$

Attacks on 6 round Feistel If G is a generator of 6 round Feistel permutations of B_{2n}, we have found an attack (described below) that uses a few (i.e. $\mathcal{O}(1)$) permutations from the generator G, $\mathcal{O}(2^{2n})$ computations and about $\mathcal{O}(2^{2n})$ random queries. So this attack has a complexity much smaller than the exhaustive search in $2^{63n \cdot 2^n}$. However since a permutation of B_{2n} has only 2^{2n} possible inputs, this attack has no real interest against a single specific 6 round Feistel scheme used in encryption.

It is interesting only if a few 6 round Feistel schemes are used. This can be particularly interesting for some cryptographic schemes using many permutations on a relatively small number of bits. For example in the Graph Isomorphism authentication scheme many permutations on about 2^{14} points are used (thus $n = 7$), or in the Permuted Kernel Problem PKP of Adi Shamir many permutations on about 2^6 points ($n = 3$ here). Then, we will be able to distinguish these permutations from truly random permutations with a small complexity if a 6 round Feistel scheme generator is used. And this, whatever the size of the secret key used in the generator may be. So we do not recommend to generate small pseudorandom permutations from 6 round Feistel schemes.

The Attack:

Let $[L_i, R_i]$ be an element of I_{2n}.
Let $\Psi^6[L_i, R_i] = [S_i, T_i]$. The attack proceeds as follows:

Step 1.

We choose specific permutation $f = G_K$.
We generate m values $f[L_i, R_i] = [S_i, T_i]$, $1 \leq i \leq m$ with the random $[L_i, R_i] \in I_{2n}$ and with $m = \mathcal{O}(2^{2n})$.

Remark: Since $m = \mathcal{O}(2^{2n})$, we cover here almost all the possible inputs $[L_i, R_i]$ for this specific permutation f.

Step 2.
We look if among these values we can find 4 pairwise distinct indices denoted by $1, 2, 3, 4$ such that these 8 equations are satisfied:

$$(\#) \begin{cases} R_1 = R_3 \\ R_2 = R_4 \\ S_1 = S_2 \\ S_3 = S_4 \\ L_1 \oplus L_3 = L_2 \oplus L_4 \\ L_1 \oplus L_3 = S_1 \oplus S_3 \\ T_1 \oplus T_2 = T_3 \oplus T_4 \\ T_1 \oplus T_2 = R_1 \oplus R_2 \end{cases}$$

(and with $R_2 \neq R_1$, $S_3 \neq S_1$ and $T_1 \neq T_2$).

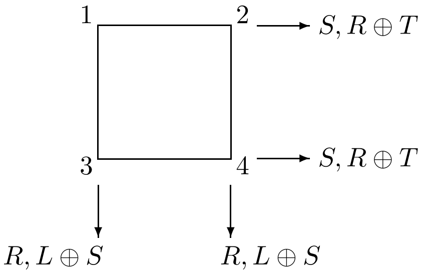

Figure 3: A representation of the 8 equations # in L, S, R, T.

It is also possible to show that all the indices that satisfy these equations can be found in $\mathcal{O}(m)$ and with $\mathcal{O}(m)$ of memory. We count the number of solutions found.

Step 3.
We try again at Step 1 with another $f = G_{K'}$ and we will do this a few times, say λ times with $\lambda = \mathcal{O}(1)$. Let α be the total number of solutions found at Step 2 for all the λ functions tested. It is possible to prove that for a generator of pseudorandom permutation of B_{2n} we have

$$\alpha \simeq \frac{\lambda m^4}{2^{8n}}.$$

Moreover it is possible to prove that for a generator of 6 round Feistel schemes the average value we get for α is

$$\alpha \geq \quad \text{about} \quad \frac{2\lambda m^4}{2^{8n}}.$$

Proof.
The proof is very similar to the proof we did for Ψ^5 (due to the lack of space we do not explicit it here).

So by counting this value α we will distinguish 6 round Feistel generators from truly random permutation generators each time when $\frac{\lambda m^4}{2^{8n}}$ is not negligible, for example when $\lambda = \mathcal{O}(1)$ and $m = \mathcal{O}(2^{2n})$, as claimed.

Examples: Thus we are able, to distinguish between a few 6 round Feistel permutations taken from a generator, and a set of truly random permutations (or from a set of random permutations with an even signature) from 32 bits to 32, within approximately 2^{32} computations and 2^{32} chosen plaintexts.

9 An attack on k round Feistel Generators

It is also possible to extend these attacks on more than 6 rounds, to any number of rounds k. However for more than 6 rounds, as already for 6 rounds, all our attacks require a complexity and a number of queries $\geq \mathcal{O}(2^{2n})$, so they can be interesting to attack generators of permutations, but not to attack a single permutation (the probability of success against one single permutation is generally negligible, and we need a few, or many permutations from the generator, in order to be able to distinguish the generator from a truly random permutation generator).

Example of attack on a Feistel generator with k rounds. Let k be an integer. For simplicity we will assume that k is even (the proof is very similar when k is odd). Let $\lambda = \frac{k}{2} - 1$. Let G be a generator of Feistel permutations of k rounds of B_{2n}. We will consider an attack with a set of equations in (L, R, S, T) illustrated in figure 3. For simplicity we do not write all the equations explicitly.

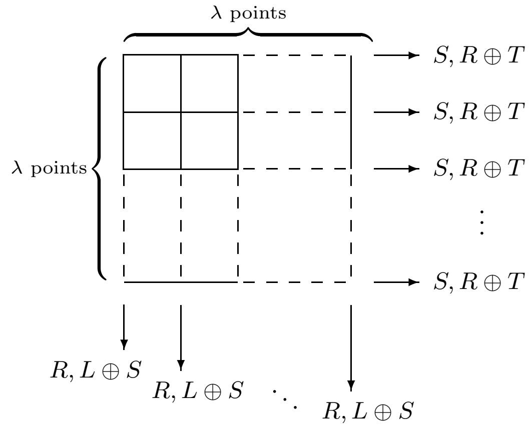

Figure 4: Modelling the $4 \cdot \lambda(\lambda - 1)$ equations in L, R, S, T.

Here we have $\mu = \lambda^2 = (\frac{k}{2} - 1)^2$ indices, and we have $4\lambda(\lambda - 1) = k^2 - 6k + 8$ equations in L, R, S, T. Here it is possible to prove that the probability that the $4\lambda(\lambda - 1)$ equations of figure 3 exist, will be about twice for a Feistel scheme with k rounds, than for a truly random permutation.

Thus, on a fixed permutation this attack succeeds with a probability in

$$\mathcal{O}\left(\frac{m^{(\frac{k}{2}-1)^2}}{2^{n \cdot 4\lambda(\lambda-1)}}\right)$$

If we take $m = \mathcal{O}(2^{2n})$ for such a permutation, it gives a probability of success in

$$\mathcal{O}\left(\frac{2^{2n(\frac{k}{2}-1)^2}}{2^{n \cdot (k^2-6k+8)}}\right)$$

So we will use $\mathcal{O}(2^{n(\frac{k^2}{2}-4k+6)})$ permutations, and the total complexity and the total number of queries on all these permutations will be $\mathcal{O}(2^{n(\frac{k^2}{2}-4k+8)})$. The total memory will be $\mathcal{O}(2^{2n})$.

Examples:

- With $k = 6$ this attack uses $\mathcal{O}(1)$ permutations and $\mathcal{O}(2^{2n})$ computations (exactly as we did in section 8).
- With $k = 8$ we need $\mathcal{O}(2^{6n})$ permutations and $\mathcal{O}(2^{8n})$ computations.

10 Conclusion

Up till now, generic attacks on Feistel schemes were known only for 1,2,3 or 4 rounds. In this paper we have seen that some generic attacks also do exist on 5 round Feistel schemes. So we do not recommend to use 5 round Feistel schemes in cryptography for general purposes. Our first attack requires $\mathcal{O}(2^{\frac{7n}{4}})$ **random** plaintext/ciphertext pairs and the same amount of computation time. Our second attack requires $\mathcal{O}(2^{\frac{3n}{2}})$ **chosen** plaintext/ciphertext pairs and the same amount of computation time. For example, it is possible to distinguish most of 5 round Feistel ciphers with blocks of 64 bits, from a random permutation from 64 bits to 64 bits, within about 2^{48} chosen queries and 2^{48} computations.

We have also seen that when we have to generate several small pseudo-random permutations we do not recommend to use a Feistel scheme generator with only 6 rounds (whatever the length of the secret key may be). As an example, it is possible to distinguish most generators of 6 round Feistel permutations from truly random permutations on 32 bits, within approximately 2^{32} computations and 2^{32} chosen plaintexts (and this whatever the length of the secret key may be).

Similar attacks can be generalised for any number of rounds k, but they require to analyse much more permutations and they have a larger complexity when k increases.

11 Acknowledgments

I would like to thank Jean-Jacques Quisquater who allowed me to do this work, as it has been done during my invited stay at the university of Louvain-La-Neuve. I also would like to thank the anonymous referee of Asiacrypt'2001, for pointing out the references [2, 3], and for observing that my attack against 5 round Feistel schemes will not in general apply as it is, against some specific round functions such as permutations. Finally I would like to thank Nicolas Courtois for his help writing this paper.

References

1. William Aiollo, Ramarathnam Venkatesan: *Foiling Birthday Attacks in Length-Doubling Transformations - Benes: A Non-Reversible Alternative to Feistel.* Eurocrypt 96, LLNCS 1070, Springer-Verlag, pp. 307-320.
2. L.R. Knudsen: *DEAL - A 128-bit Block Cipher*, Technical report #151, University of Bergen, Department of Informatics, Norway, February 1998. Submitted as a candidate for the Advanced Encryption Standard. Available at http://www.ii.uib.no/∼larsr/newblock.html
3. L.R. Knudsen, V. Rijmen: *On the Decorrelated Fast Cipher (DFC) and its Theory.* Fast Software Encryption (FSE'99), Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636, pp. 81-94, Springer, 1999.
4. M. Luby, C. Rackoff, *How to construct pseudorandom permutations from pseudorandom functions*, SIAM Journal on Computing, vol. 17, n. 2, pp. 373-386, April 1988.
5. Moni Naor and Omer Reingold, *On the construction of pseudo-random permutations: Luby-Rackoff revisited*, J. of Cryptology, vol 12, 1999, pp. 29-66. Extended abstract in: Proc. 29th Ann. ACM Symp. on Theory of Computing, 1997, pp. 189-199.
6. J. Patarin, *Pseudorandom Permutations based on the DES Scheme*, Eurocode'90, LNCS 514, Springer-Verlag, pp. 193-204.
7. J. Patarin, *New results on pseudorandom permutation generators based on the DES scheme*, Crypto'91, Springer-Verlag, pp. 301-312.
8. J. Patarin *About Feistel Schemes with Six (or More) Rounds*, in Fast Software Encryption 1998, pp. 103-121.